CSCI 361: Reading Assignment # 11

Your Name: Noah Cage

Due: Tuesday, Nov 5, in class

Assigned reading for this lecture:

- Sipser Third Edition Chap 5.1

(Reduction via Computation Histories) To prove that some properties of the language of CFGs such as ALL_{CFG} (Is the language of a given CFG contain all strings) and EQ_{CFG} (do two CFGs have the same language) we need a new reduction technique that uses the computation histories of TMs. Read **Definition 5.5** on Page 221 and the proof of Theorem 5.13 on Page 225.

Questions:

1. Consider the grammar G in the proof. If M does not accept w then what is its language?

If M does not accept w then $L(G) = \Sigma^*$

2. Why does the PDA alternate between a string that is a TM configuration and a string that is the reverse of TM configuration?

The C_i^{th} and C_{i+1}^{th} configurations need to be compared. With a stack we can compare W and W^R but cannot compare W and W.

The proper configuration for C_i and C_{i+1} are

$C_i \# C_i^R$ transition #

Thus w/ a PDA C_i and C_{i+1} can be compared.

CSCI 361: Reading Assignment # 11 Your Name: Kunal Pal

Due: Tuesday, Nov 5, in class

Assigned reading for this lecture:

- Sipser Third Edition Chap 5.1

(Reduction via Computation Histories) To prove that some properties of the language of CFGs such as ALL_{CFG} (Is the language of a given CFG contain all strings) and EQ_{CFG} (do two CFGs have the same language) we need a new reduction technique that uses the computation histories of TMs. Read **Definition 5.5** on Page 221 and the proof of Theorem 5.13 on Page 225.

Questions:

1. Consider the grammar G in the proof. If M does not accept w then what is its language?

If M does not accept w, its grammar G would generate all strings that fail to be an accepting computation history for M on w. In this case, it would end up being all possible strings over its alphabet.

2. Why does the PDA alternate between a string that is a TM configuration and a string that is the reverse of TM configuration?

The PDA alternates between a string that is a TM configuration and a string that is the reverse of the TM configuration to align the symbol orders whenever we push/pop from the stack, allowing us to compare adjacent configurations correctly.

CSCI 361: Reading Assignment # 11 Your Name: Juan Mendez

Due: Tuesday, Nov 5, in class

Assigned reading for this lecture:

- Sipser Third Edition Chap 5.1

(Reduction via Computation Histories) To prove that some properties of the language of CFGs such as ALL_{CFG} (Is the language of a given CFG contain all strings) and EQ_{CFG} (do two CFGs have the same language) we need a new reduction technique that uses the computation histories of TMs. Read **Definition 5.5** on Page 221 and the proof of Theorem 5.13 on Page 225.

Questions:

1. Consider the grammar G in the proof. If M does not accept w then what is its language?

If M does not accept w, then $L(G) = \Sigma^*$.

2. Why does the PDA alternate between a string that is a TM configuration and a string that is the reverse of TM configuration?

This allows the PDA to properly compare configurations when pushing/popping them off the stack.

CSCI 361: Reading Assignment # 11

Your Name: Prairie

Due: Tuesday, Nov 5, in class

Assigned reading for this lecture:

- Sipser Third Edition Chap 5.1

(Reduction via Computation Histories) To prove that some properties of the language of CFGs such as ALL_{CFG} (Is the language of a given CFG contain all strings) and EQ_{CFG} (do two CFGs have the same language) we need a new reduction technique that uses the computation histories of TMs. Read **Definition 5.5** on Page 221 and the proof of Theorem 5.13 on Page 225.

Questions:

1. Consider the grammar G in the proof. If M does not accept w then what is its language?

$L(M) = \emptyset$

No accepting computation history exists, so all strings must fail somehow

2. Why does the PDA alternate between a string that is a TM configuration and a string that is the reverse of TM configuration?

So that when the strings are popped off the stack, they can be in an order allowing for comparison w/ C_{i+1}

CSCI 361: Reading Assignment # 11

Your Name: Jarin S.

Due: Tuesday, Nov 5, in class

Assigned reading for this lecture:

- Sipser Third Edition Chap 5.1

(Reduction via Computation Histories) To prove that some properties of the language of CFGs such as ALL_{CFG} (Is the language of a given CFG contain all strings) and EQ_{CFG} (do two CFGs have the same language) we need a new reduction technique that uses the computation histories of TMs. Read **Definition 5.5** on Page 221 and the proof of Theorem 5.13 on Page 225.

Questions:

1. Consider the grammar G in the proof. If M does not accept w then what is its language?

All strings that are not accepting ~~configurations~~ computation histories for M on w.

2. Why does the PDA alternate between a string that is a TM configuration and a string that is the reverse of TM configuration?

The PDA alternates between a string that is a TM configuration & one that is reversed so that it is in the correct order for comparison between Ci & Ci+1.

CSCI 361: Reading Assignment # 11

Your Name: Lola Kovalski

Due: Tuesday, Nov 5, in class

Assigned reading for this lecture:

- Sipser Third Edition Chap 5.1

(Reduction via Computation Histories) To prove that some properties of the language of CFGs such as ALL_{CFG} (Is the language of a given CFG contain all strings) and EQ_{CFG} (do two CFGs have the same language) we need a new reduction technique that uses the computation histories of TMs. Read **Definition 5.5** on Page 221 and the proof of Theorem 5.13 on Page 225.

Questions:

1. Consider the grammar G in the proof. If M does not accept w then what is its language?

G generates all strings if M does not accept w

2. Why does the PDA alternate between a string that is a TM configuration and a string that is the reverse of TM configuration?

So that when we pop C_i off of the stack, we can compare to C_{i+1}

CSCI 361: Reading Assignment # 11

Your Name: Ahmed Hussain.

Due: Tuesday, Nov 5, in class

Assigned reading for this lecture:

- Sipser Third Edition Chap 5.1

(Reduction via Computation Histories) To prove that some properties of the language of CFGs such as ALL_{CFG} (Is the language of a given CFG contain all strings) and EQ_{CFG} (do two CFGs have the same language) we need a new reduction technique that uses the computation histories of TMs. Read **Definition 5.5** on Page 221 and the proof of Theorem 5.13 on Page 225.

Questions:

1. Consider the grammar *G* in the proof. If *M* does not accept *w* then what is its language?

$L(G) = \Sigma^*$

2. Why does the PDA alternate between a string that is a TM configuration and a string that is the reverse of TM configuration?

The way a stack works - LIFO -
So need w — w^R —

CSCI 361: Reading Assignment # 11

Your Name: Juskaran Singh

Due: Tuesday, Nov 5, in class

Assigned reading for this lecture:

- Sipser Third Edition Chap 5.1

(Reduction via Computation Histories) To prove that some properties of the language of CFGs such as ALL_{CFG} (Is the language of a given CFG contain all strings) and EQ_{CFG} (do two CFGs have the same language) we need a new reduction technique that uses the computation histories of TMs. Read **Definition 5.5** on Page 221 and the proof of Theorem 5.13 on Page 225.

Questions:

1. Consider the grammar *G* in the proof. If *M* does not accept *w* then what is its language?

L(G) → The language generated by G

2. Why does the PDA alternate between a string that is a TM configuration and a string that is the reverse of TM configuration?

This is so that it is able to push a configuration so that when it is popped, the order is suitable for comparison w/ the next string.

CSCI 361: Reading Assignment # 11

Your Name: Daniel Seo

Due: Tuesday, Nov 5, in class

Assigned reading for this lecture:

- Sipser Third Edition Chap 5.1

(Reduction via Computation Histories) To prove that some properties of the language of CFGs such as ALL_{CFG} (Is the language of a given CFG contain all strings) and EQ_{CFG} (do two CFGs have the same language) we need a new reduction technique that uses the computation histories of TMs. Read **Definition 5.5** on Page 221 and the proof of Theorem 5.13 on Page 225.

Questions:

1. Consider the grammar G in the proof. If M does not accept w then what is its language?

G constructs all non-valid history of computations. G must generate the entirety of Σ^*.

2. Why does the PDA alternate between a string that is a TM configuration and a string that is the reverse of TM configuration?

This is because the PDA employs a stack structure. What comes in first comes out last.

CSCI 361: Reading Assignment # 11

Your Name: Abel Mesfin

Due: Tuesday, Nov 5, in class

Assigned reading for this lecture:

- Sipser Third Edition Chap 5.1

(Reduction via Computation Histories) To prove that some properties of the language of CFGs such as ALL_{CFG} (Is the language of a given CFG contain all strings) and EQ_{CFG} (do two CFGs have the same language) we need a new reduction technique that uses the computation histories of TMs. Read **Definition 5.5** on Page 221 and the proof of Theorem 5.13 on Page 225.

Questions:

1. Consider the grammar *G* in the proof. If *M* does not accept *w* then what is its language?

$L(G) = \{\Sigma^*\}$

2. Why does the PDA alternate between a string that is a TM configuration and a string that is the reverse of TM configuration?

$\Rightarrow$ Makes it easier for comparison.
(suitable)